



THE UNIVERSITY *of* EDINBURGH

## Edinburgh Research Explorer

### Phylogenetic convolutional neural networks in metagenomics

**Citation for published version:**

Fioravanti, D, Giarratano, Y, Maggio, V, Agostinelli, C, Chierici, M, Jurman, G & Furlanello, C 2018, 'Phylogenetic convolutional neural networks in metagenomics' BMC Bioinformatics, vol. 19, no. Suppl 2, pp. 49. DOI: 10.1186/s12859-018-2033-5

**Digital Object Identifier (DOI):**

[10.1186/s12859-018-2033-5](https://doi.org/10.1186/s12859-018-2033-5)

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Publisher's PDF, also known as Version of record

**Published In:**

BMC Bioinformatics

**Publisher Rights Statement:**

© The Author(s). 2018 Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated.

**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



RESEARCH

Open Access



# Phylogenetic convolutional neural networks in metagenomics

Diego Fioravanti<sup>1,2†</sup>, Ylenia Giarratano<sup>3†</sup>, Valerio Maggio<sup>1†</sup>, Claudio Agostinelli<sup>4</sup>, Marco Chierici<sup>1</sup>, Giuseppe Jurman<sup>1\*</sup> and Cesare Furlanello<sup>1</sup>

From Bringing Maths to Life 2017 BMTL 2017  
Naples, Italy. 07-09 June 2017

## Abstract

**Background:** Convolutional Neural Networks can be effectively used only when data are endowed with an intrinsic concept of neighbourhood in the input space, as is the case of pixels in images. We introduce here Ph-CNN, a novel deep learning architecture for the classification of metagenomics data based on the Convolutional Neural Networks, with the patristic distance defined on the phylogenetic tree being used as the proximity measure. The patristic distance between variables is used together with a sparsified version of MultiDimensional Scaling to embed the phylogenetic tree in a Euclidean space.

**Results:** Ph-CNN is tested with a domain adaptation approach on synthetic data and on a metagenomics collection of gut microbiota of 38 healthy subjects and 222 Inflammatory Bowel Disease patients, divided in 6 subclasses. Classification performance is promising when compared to classical algorithms like Support Vector Machines and Random Forest and a baseline fully connected neural network, e.g. the Multi-Layer Perceptron.

**Conclusion:** Ph-CNN represents a novel deep learning approach for the classification of metagenomics data. Operatively, the algorithm has been implemented as a custom Keras layer taking care of passing to the following convolutional layer not only the data but also the ranked list of neighbourhood of each sample, thus mimicking the case of image data, transparently to the user.

**Keywords:** Metagenomics, Deep learning, Convolutional neural networks, Phylogenetic trees

## Background

Biological data is often complex, heterogeneous and hard to interpret, thus a good testbed for Deep Learning (DL) techniques [1]. The superiority of deep neural network approaches is acknowledged in a first group of biological and clinical tasks, with new results constantly flowing in the literature [2–4]. However, DL is not yet a “silver bullet” in bioinformatics; indeed a number of issues are still limiting its potential in applications, including limited data availability, result interpretation and hyperparameters tuning [5]. In particular, DL approaches has so far failed in showing an advantage in metagenomics, either

in terms of achieving better performance or detecting meaningful biomarkers. This lack of significant results led Ditzler and coauthors [6] to state that deep learning “may not be suitable for metagenomic application”; nevertheless, novel promising attempts have recently appeared [7, 8]. With a slight abuse of notation, in what follows we use the more common term metagenomics even in the 16S metabarcoding case, following the notation of the MetaHIT paper [9] and the official Illumina documentation [10].

Unique among other omics, metagenomics features are endowed with a hierarchical structure provided by the phylogenetic tree defining the bacterial clades. In detail, samples are usually described by features called Operational Taxonomic Units (OTU). For each OTU, its position as a leaf of the phylogenetic tree and its abundance value in the sample are automatically extracted by

\*Correspondence: [jurman@fbk.eu](mailto:jurman@fbk.eu)

†Equal contributors

<sup>1</sup>Fondazione Bruno Kessler (FBK), Via Sommarive 18 Povo, I-38123 Trento, Italy  
Full list of author information is available at the end of the article

bioinformatics analysis. In this work we exploit this hierarchical structure as an additional information for the learning machine to better support the profiling process: this has been proposed before in [11, 12], but only in shallow learning contexts, to support classification or for feature selection purposes. We aim to exploit the phylogenetic structure to enable adopting the Convolutional Neural Network (CNN) DL architecture otherwise not useful for omics data: we name this novel solution *Ph-CNN*. Indeed CNNs are the elective DL method for image classification [13, 14] and they work by convolving subsets of the input image with different filters. The operation is based on the matricial structure of a digital image and, in particular, the concept of neighbours of a given pixel. Using the same architecture for non-image data requires the availability of an analogous proximity measure between features.

In the metagenomics case, such measure can be inherited by the tree structure connecting the OTUs and the neighbourhood are naturally defined once an appropriate tree distance between two OTUs is defined. In this paper, we adopt the patristic distance, i.e., the sum of the lengths of all branches connecting two OTUs on the phylogenetic tree [15]. By definition, the output of a CNN consists of linear combinations of the original input features: this implies that, if *Ph-CNN* includes more CNN layers, the problem of finding the neighbours of a OTU is shifted into the hardest task of finding the neighbours of a linear combination of OTUs. The workaround here is mapping OTUs into points of a  $k$ -dimensional metric space preserving distances as well as possible via a MultiDimensional Scaling (MDS) projection [16]: the use of MDS is allowed because the patristic distance is Euclidean [17]. A further refinement is provided by sparsifying MDS via regularized low rank matrix approximation [18] through the addition of the smoothly clipped absolute deviation penalty [19], tuned by cross-validation. A caveat: different topologies of the phylogenetic tree lead to different distance matrices. As pointed out in [20], different softwares can produce very different topologies, thus the choice of the software and its version in the whole metagenomic pipeline play a critical role here as a relevant source of variability, and this is true for all the steps throughout the whole preprocessing workflow.

The convolutional layer combined with the neighbours detection algorithm is operatively implemented as a novel Keras layer [21] called *Phylo-Conv*. *Ph-CNN* consists of a stack of *Phylo-Conv* layers first flattened then terminating with a Fully Connected (Dense) and a final classification layer. The experimental setup is realized as a 10x5-fold cross-validation schema with a feature selection and ranking procedure, implementing the Data Analysis Protocol (DAP) developed within the US-FDA led initiatives MAQC/SEQC [22, 23], to control for selection bias and other overfitting effects and warranting honest

performance estimates on external validation data subsets. Top ranking features are recursively selected as the  $k$ -best at each round, and finally aggregated via Borda algorithm [24]. Model performance is computed for increasing number of best ranking features by Matthews Correlation Coefficient (MCC), the measure that better convey in an unique value the confusion matrix of a classification task, even in the multiclass case [25–27]. Experiments with randomized features and labels are also performed as model sanity check.

We demonstrate *Ph-CNN* characteristics with experiments on both synthetic and real omics data. For the latter type, we consider Sokol's lab data [28] of microbiome information for 38 healthy subjects (HS) and 222 inflammatory bowel disease (IBD) patients. The bacterial composition was analysed using 16S sequencing and a total number of 306 different OTUs was found. IBD is a complex disease arising as a result of the interaction of environmental and genetic factors inducing immunological responses and inflammation in the intestine and primarily including ulcerative colitis (UC) and Crohn's disease (CD). Both disease classes are characterized by two conditions: flare (f), when symptoms reappear or worsen, and remission (r), when symptoms are reduced or disappear. Finally, since CD can affect different parts of the intestine, we distinguish ileal Crohn's disease (iCD) and colon Crohn's disease (cCD). Note however that the number of non zero features varies for the different tasks, (defined by disease, condition site) since some features may vanish on all samples of a class.

Synthetic data are constructed mimicking the structure of the IBD dataset. They are generated as compositional data from multivariate normal distributions with given covariances and means: in particular, to provide different complexity levels in the classification task, four different instances of data are generated with different ratios between class means. On both data types, the *Ph-CNN* architecture than compared with state-of-art shallow algorithms as Support Vector Machines (SVMs) and Random Forest (RF), and with alternative neural networks methods such as Multi-Layer Perceptron (MLPNN).

Moreover, the bacterial genera detected as top discriminating features are consistent with the key players known in the literature to play a major role during the IBD progression. Since the direct use of *Ph-CNN* on the IBD dataset leads to overfitting after few epochs due to the small sample size, the IBD dataset is used in a transfer learning (domain adaptation) task.

Finally, although described and demonstrated on bacterial metagenomics, *Ph-CNN* can be applied to every metagenomics datasets whose features are associated to a taxonomy and thus to a tree structure, as in the case of metagenomics of relatively large eukaryotes now appearing in the literature [29].

A preliminary version of the method has been presented as the M.Sc. thesis [30].

## Methods

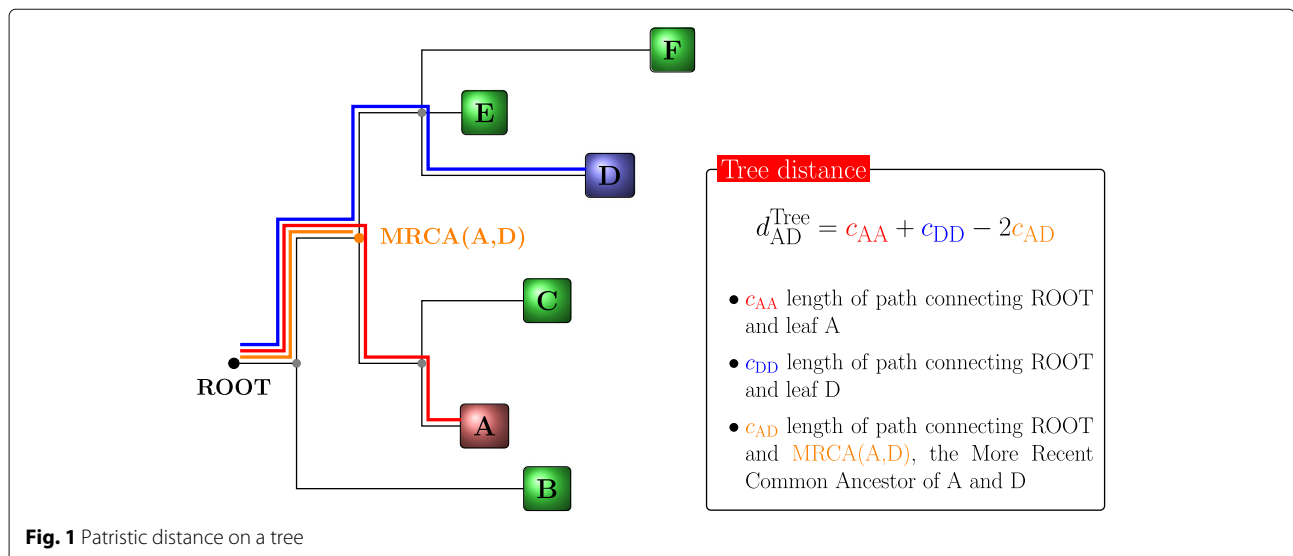
### Ph-CNN

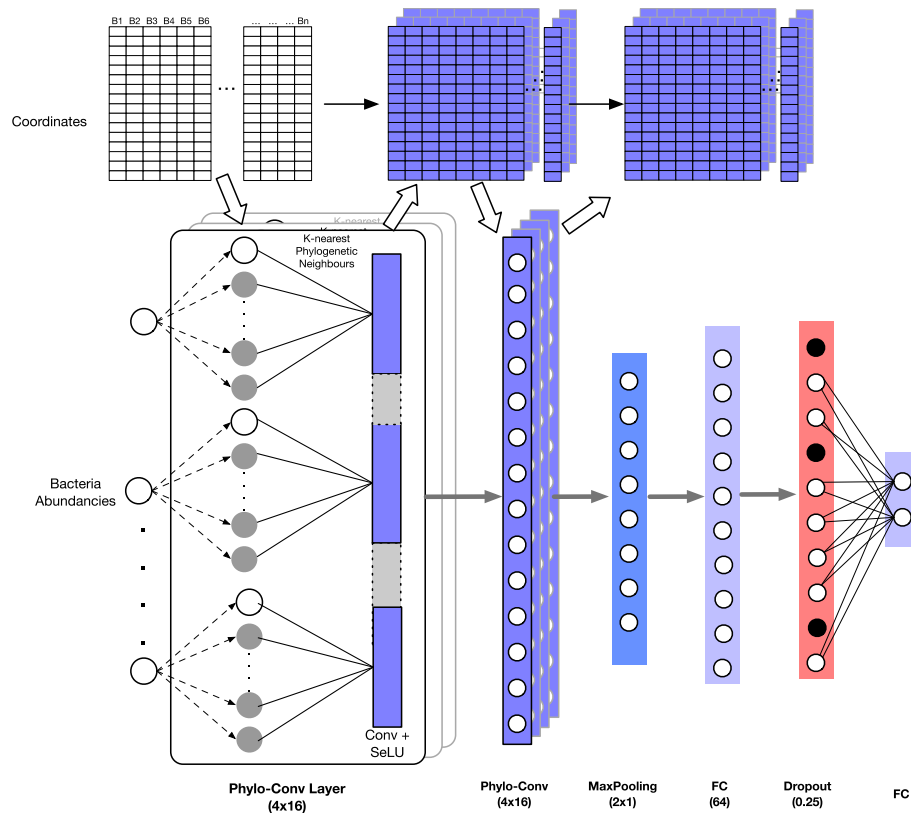
The Ph-CNN is a novel DL architecture aimed at effectively including the phylogenetic structure of metagenomics data into the learning process. In detail, Ph-CNN takes as input both the OTU abundances table and the OTU distance matrix described hereafter and provides as output the class of each sample. The core of the network is the Phylo-Conv layer, a novel Keras [21] layer coupling convolution with the neighbours detection. In a generic Phylo-Conv layer, the structure input is represented by a collection of meta-leaves, i.e. linear combinations of the leaves of the original tree; for the first Phylo-Conv layer, the structure input is simply the original set of leaves (OTUs, in the metagenomic case). The neighbour detection procedure identifies the  $k$ -nearest neighbours of a given metaleaf: the linear combination of the abundances of the corresponding OTUs is then convolved with the filters by the CNN. The core ingredient is the choice of a metric on the phylogenetic tree [31, 32] quantifying the distance between two leaves on the tree. In the current case, we choose the patristic distance [15], i.e., the sum of the lengths of all branches connecting two OTUs. In Fig. 1 we show how to compute the patristic distance between two leaves in a tree. To deal with the problem of finding neighbours for linear combinations of leaves, we map the discrete space of the set of leaves into an Euclidean space of a priori chosen dimension, by associating each leaf to a point  $P_i$  in the Euclidean space with variable Euclidean coordinates preserving the tree distance as well as possible. The algorithm used for this mapping is the metric Multidimensional Scaling (MDS)

[16], whose use is allowed because the square root  $\sqrt{d^{\text{Tree}}}$  of the patristic distance in Fig. 1 is euclidean [17], that is, the matrix  $(P_i \cdot P_j)$  is positive semidefinite. Thus, given a linear combination of OTUs, it is possible to compute its  $k$ -nearest neighbours as the  $k$ -nearest neighbours of the corresponding linear combination of projected points  $P_i$ : in all experiments, the number of neighbours  $k$  is set to 16. The selected neighbours are then convolved with the 16 filters on the CNN. The Phylo-Conv is then repeated; finally, the terminating layers of the Ph-CNN are a Max-Pooling, then a Flatten layer and, finally, a Fully Connected with 64 neurons (changed to 128 for the transfer learning experiments) and a 0.25 Dropout. Each convolutional layer has a Scaled Exponential Linear Units (SELU) [33] as the activation function and the dense layer in transfer learning experiments uses a sigmoid activation function. Adam [34] is used as optimizer with learning rate 0.0005.

### Experimental setup

To ensure predictive power and limit overfitting effect, the experimental framework is structured following the guidelines recommended by the US-FDA led studies MAQC/SEQC [22, 23] that investigated the development of predictive models for the analysis of high-throughput data. In particular, the Ph-CNN (shown in Fig. 2) becomes the core of an experimental setup designed according to the DAP shown in Fig. 3, based on 10 repetitions of a 5-fold cross validation. In detail, the dataset is first partitioned into a non overlapping training set and test set, preserving the original stratification, i.e., the ratio between sample size across classes. In the experiments described hereafter, the training set size is 80% of the original dataset. Then the training set undergoes 10 rounds of 5-fold stratified cross validation, with Ph-CNN as the classifier and  $k$ -Best as the feature selection

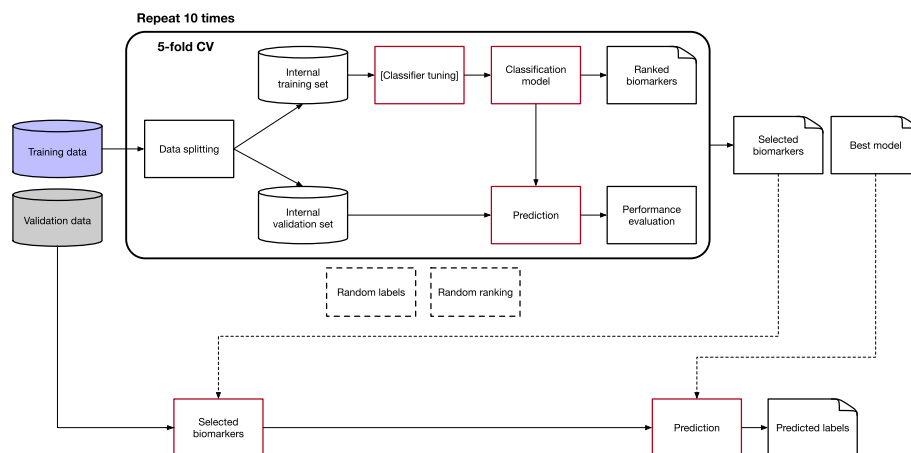




**Fig. 2** The structure of Ph-CNN. In this configuration, Ph-CNN is composed by two PhyloConv layers followed by a Fully Connected layer before decision

algorithm, with ANOVA F-value as the ranking score. At each round, several models are built for increasing number of ranked features (in this case, 25%, 50%, 75% and 100% of the total features) using Matthews Correlation Coefficient (MCC) [25, 26] as the performance measure. MCC is rated as an elective choice [22, 23] for effectively combining into a single figure the confusion matrix of a classification task, and hence for evaluating

classifiers' outcomes even when classes are imbalanced. Originally designed for binary discrimination, a multi-class version has also been developed [27, 35]. MCC values range between -1 and 1, where 1 indicates perfect classification, -1 perfect misclassification and 0 for coin tossing or attribution of every samples to the largest class. The lists of ranked features produced within the cross-validation schema are then fused into a single



**Fig. 3** Data Analysis Protocol for the experimental framework



ranked list using the Borda method [36–38]. The subset of the fused list of ranked featured corresponding to the higher MCC value is selected as the optimal set of discriminating features for the classification tasks. The fused list is further used to build the models for increasing number of features on the validation set (sometimes called the external validation set, to avoid ambiguities with the internal validation sets created at each CV round). Finally, as sanity check for the procedure, the same methodology is applied several times on instances of the original dataset after randomly permuting the labels (random labels in Fig. 3) and picking up random features instead of selecting them on the basis of the model performances (random features in Fig. 3); in both cases, a procedure unaffected by systematic bias should return an average MCC close to 0.

### The IBD dataset

The IBD dataset has been originally published in [28] for a study aimed at investigating correlation between bacteria and fungal microbiota in different stages of Inflammatory Bowel Disease. IBD is a clinical umbrella term defining a group of inflammatory conditions of the digestive tract, induced by the interactions of environmental and genetic factors leading to immunological responses and inflammation in the intestine: Ulcerative colitis (UC) and Crohn's disease (CD) are the two main conditions. The onset of bacterial dysbiosis of the gut microbiota has recently been observed in patients affected by IBD: a decrease in the abundance of *Firmicutes* phylum and an increase for *Proteobacteria* phylum, albeit the exact pathogenesis of IBD remains unknown [39, 40].

The IBD dataset includes both fungal and bacterial abundances from faecal samples of 38 healthy subjects (HS) and 222 IBD patient, collected at the Gastroenterology Department of the Saint Antoine Hospital (Paris, France). In the present study, we only consider the bacterial data subset on which we have a deeper analysis experience.

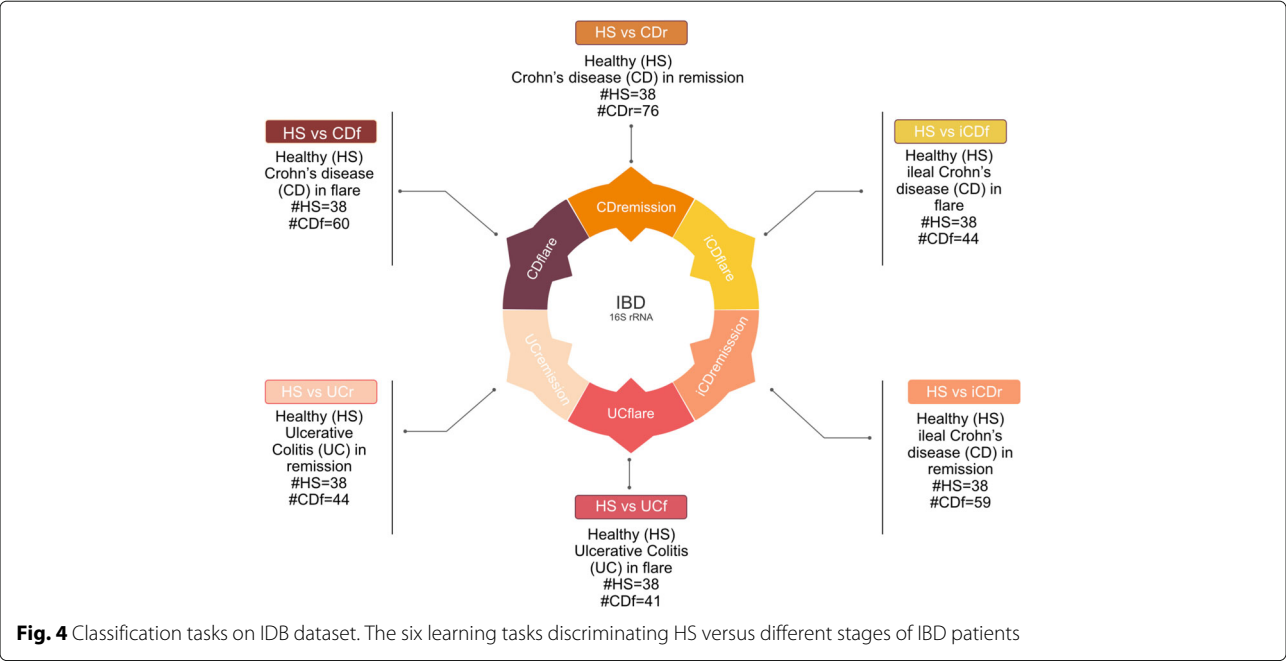
IBD patients are divided in two classes according to the disease phenotype UC and CD. Each disease class is further characterized by two conditions: flare (f), if symptoms reappear or worsen, and remission (r), if symptoms are reduced or disappear. Moreover, since CD can affected different parts of the intestine we further partition the data subset into ileal Crohn's disease (iCD) and colon Crohn's disease (cCD). In Table 1 we summarize the sample distribution. In terms of learning tasks, we investigate the six classification tasks discriminating HS versus the six IBD partitions UCf, UCr, CDf, CDr, iCDf and iCDr, as graphically shown in Fig. 4.

The bacterial composition is analysed using 16S rRNA sequencing, demultiplexed and quality filtered using the QIIME 1.8.0 software [41, 42]; minimal sequence length was 200pb. Sequences are assigned to OTUs using the

**Table 1** Patient stratification in the IBD dataset

HS	IBD patients					
	CDf		CDr		UCf	UCr
	iCDf	cCDf	iCDr	cCDr		
38	44	16	59	18	41	44
14.6%	16.9%	6.1%	22.7%	6.9%	15.8%	16.9%

UCLUST [43] algorithm with 97% threshold pairwise identity and taxonomically classified using Greengenes reference database [44]. Samples with less than 10,000 sequences are excluded from analysis. The number of different OTUs found is 306: each OTU in the data sets is associated to the sequences with the same taxonomy. Among those sequences, the one with the highest median abundance across samples is chosen as the OTU representative. Since many sequences are not in the Greengenes database, OTUs can have an unassigned taxonomy: in this case, the OTU is removed from the analysis. The actual number of OTUs used in the analyses is 259: for some discrimination tasks, however, the number of features is smaller, since some of them are all zeros for all samples in a class. The distance between the OTUs is inferred first by aligning sequences using the NAST algorithm [45, 46] and then by building the phylogenetic tree via the RAxML algorithm [47]. In detail, RAxML has been used in the rapid bootstrap mode with 100 runs, searching for bestscoring Maximum Likelihood tree (best tree). No statistical filter has been applied to the node/edge quality value of the obtained tree. Low supported branches are used as they appear in the RAxML best tree output. The phylogenetic tree for the IBD dataset resulting from the described procedure is shown in Fig. 5: largest abundance values of gut microbiota belong to *Firmicutes* (red), *Bacteroidetes* (green) and *Proteobacteria* (blue), consistently with the published literature. As pointed out already, uncertainties in topology may create fake distances which will ultimately negatively affect all downstream analyses, with software variability playing a major role [20]. While our choice here is to follow the processing pipeline in [28] to ensure data reproducibility, a stronger support in building the phylogenetic tree can be obtained by using alternative algorithms, such as the maximum-likelihood nearest-neighbor interchanges implemented in FastTree2 [48]. Analogous considerations can be formulated for all steps of the preprocessing pipeline: for instance, QIIME is now at version 1.9.1, with major release QIIME 2 scheduled for January 2018. Moreover, the Greengenes database is actually outdated, so switching to another reference database, such as SILVA [49], for the OTU definition would improve the reliability of the process. Finally, the choice to exclude taxonomically unclassified sequences from successive analysis is



arbitrary: excluding OTU sequences after a chimera-removal procedure would result in a more precise set of OTUs.

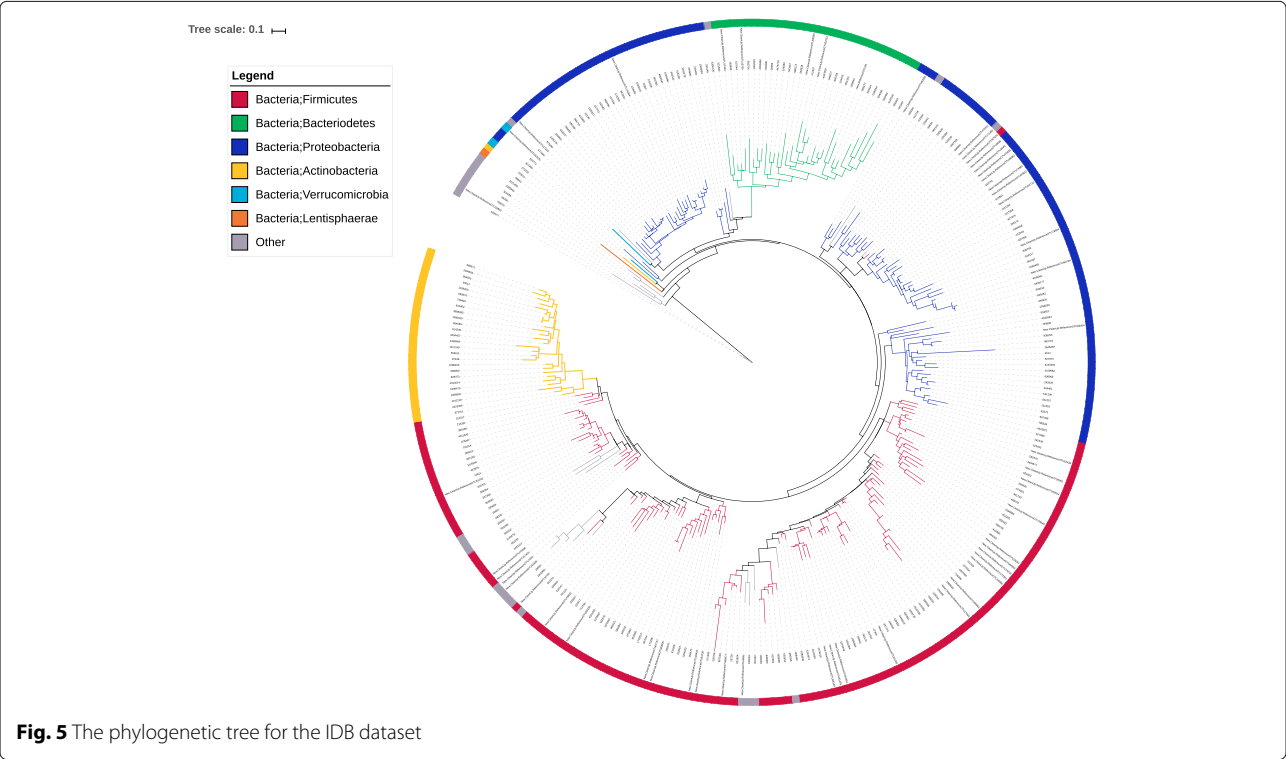
**The synthetic datasets**

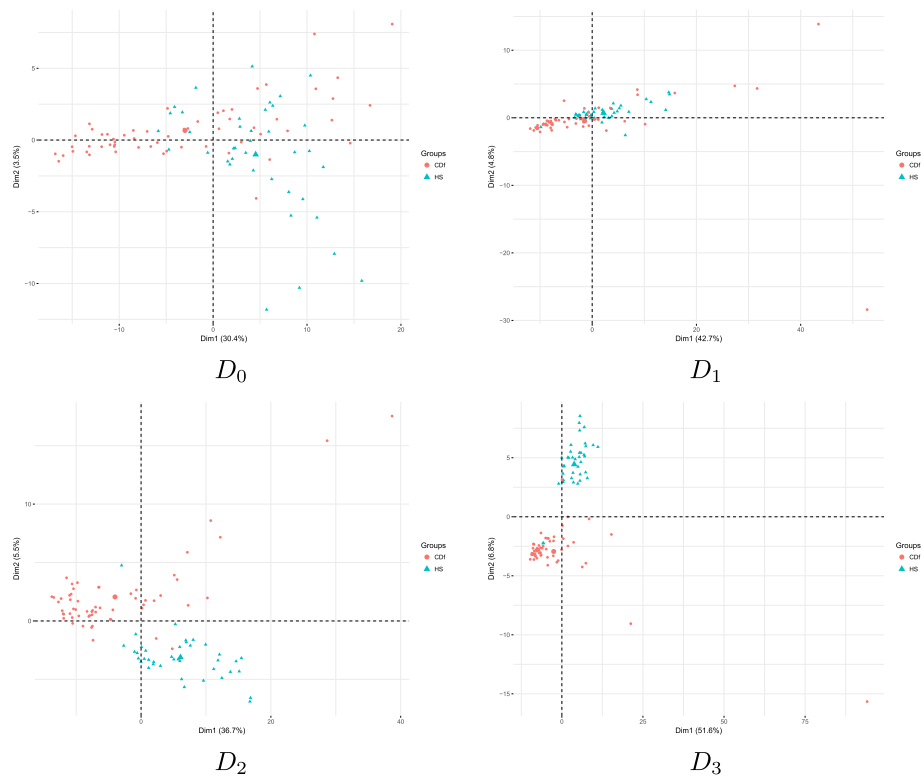
The synthetic datasets are generated as compositional data, i.e., vectors lying in the  $p$ -dim Aitchison simplex

$$\mathcal{S} = \left\{ \mathbf{x} = (x_1, x_2, \dots, x_p) \in (\mathbb{R}_0^+)^p \text{ with } \sum_{j=1}^p x_j = 1 \right\},$$

whose structure resembles the IBD data.

Note that the application of standard multivariate statistical procedures on compositional data requires adopting adequate invertible transformation procedures to preserve the constant sum constrain [50]: a standard map is the isometric log ratio *ilr* [51], projecting





**Fig. 6** Principal component analysis for the 4 synthetic datasets  $D_0, D_1, D_2, D_3$ , with same sample sizes as in the IBD dataset. Larger values of  $\alpha$  correspond to more separate classes HR and CDF

**Table 2** Dataset **D**: classification performance of Ph-CNN compared to other classifiers on Healthy vs. UCF classification task

UCf	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
63	0.794	0.785	0.803	0.799	0.793	0.803
125	0.852	0.845	0.860	0.861	0.857	0.865
188	0.920	0.916	0.925	0.924	0.921	0.926
250	0.940	0.937	0.944	0.943	0.941	0.945
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
63	0.701	0.692	0.721	0.729	0.723	0.736
125	0.838	0.834	0.842	0.843	0.837	0.849
188	0.865	0.861	0.869	0.902	0.899	0.906
250	0.898	0.894	0.901	0.903	0.900	0.907

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

**Table 3** Dataset **D**: classification performance of Ph-CNN compared to other classifiers on Healthy vs. UCr classification task

UCr	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
60	0.861	0.855	0.867	0.811	0.807	0.815
119	0.893	0.888	0.899	0.866	0.862	0.870
178	0.906	0.900	0.911	0.892	0.888	0.895
237	0.920	0.916	0.924	0.917	0.914	0.920
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
60	0.873	0.869	0.443	0.797	0.792	0.801
119	0.877	0.873	0.877	0.799	0.794	0.803
178	0.859	0.855	0.880	0.791	0.787	0.794
237	0.849	0.844	0.854	0.790	0.786	0.795

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)



**Table 4** Dataset **D**: classification performance of Ph-CNN compared to other classifiers on Healthy vs. Cdf classification task

Cdf	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.785	0.775	0.795	0.781	0.776	0.785
130	0.832	0.825	0.840	0.833	0.829	0.838
195	0.896	0.891	0.901	0.910	0.907	0.912
259	0.927	0.924	0.930	0.920	0.918	0.923
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.604	0.593	0.614	0.764	0.760	0.769
130	0.821	0.817	0.825	0.805	0.800	0.810
195	0.830	0.825	0.836	0.863	0.860	0.867
259	0.858	0.854	0.862	0.880	0.877	0.883

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

the  $p$ -dimensional Aitchison simplex isometrically to a  $p - 1$ -dimensional euclidian vector. Transformations like *ilr* allow using unconstrained statistics on the transformed data, with inferences mapped back to original compositional data through the inverse map.

The construction of the synthetic data starts from the IDB dataset, and in particular from the two subsets of the HS and Cdf samples (by abuse of notation, we use the same identifier for both the class and the compositional data subset). Classes HS and Cdf are defined by

**Table 5** Dataset **D**: classification performance of Ph-CNN compared to other classifiers on Healthy vs. CDr classification task

CDr	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.714	0.705	0.723	0.740	0.734	0.746
129	0.799	0.793	0.806	0.802	0.798	0.808
193	0.850	0.844	0.856	0.860	0.857	0.864
257	0.890	0.884	0.895	0.880	0.877	0.882
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.498	0.473	0.521	0.688	0.682	0.695
129	0.783	0.778	0.788	0.744	0.740	0.784
193	0.766	0.759	0.773	0.762	0.756	0.767
257	0.788	0.782	0.794	0.765	0.761	0.771

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

**Table 6** Dataset **D**: classification performance of Ph-CNN compared to other classifiers on Healthy vs. iCdf classification task

iCdf	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
62	0.781	0.772	0.790	0.804	0.799	0.808
124	0.863	0.854	0.871	0.861	0.858	0.865
186	0.922	0.918	0.926	0.921	0.919	0.924
247	0.944	0.941	0.947	0.941	0.939	0.942
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
62	0.845	0.840	0.849	0.748	0.743	0.753
124	0.889	0.886	0.893	0.808	0.803	0.814
186	0.879	0.875	0.883	0.880	0.877	0.883
247	0.901	0.899	0.904	0.890	0.887	0.893

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

259 features (OTU), and they include 38 and 60 samples respectively. The key step is the generation of the synthetic  $HS_s^\alpha$  and  $CDf_s^\alpha$  subsets, sampled from multivariate normal distributions with given covariances and mean.

Operatively, let  $HS'$  and  $CDf'$  the *ilr*-transformed HS and Cdf subsets. Then compute the featurewise mean  $\mu(HS') = (\mu_1(HS'), \mu_2(HS'), \dots, \mu_{258}(HS'))$  and  $\Sigma(HS')$  the covariance matrix. Analogously compute  $\mu(CDf')$  and  $\Sigma(CDf')$ . Consider now the matrix  $HS'_0$  defined by subtracting to each row of  $HS'$  the vector of the means:  $(HS'_0)_i = (HS')_i - \mu(HS')$ , and define analogously the

**Table 7** Dataset **D**: classification performance of Ph-CNN compared to other classifiers on Healthy vs. iCDr classification task

iCDr	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.753	0.744	0.763	0.773	0.769	0.779
129	0.830	0.823	0.837	0.834	0.830	0.837
193	0.884	0.878	0.889	0.893	0.891	0.896
257	0.910	0.905	0.915	0.907	0.904	0.909
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
63	0.807	0.802	0.812	0.724	0.719	0.729
125	0.822	0.816	0.827	0.794	0.788	0.800
188	0.831	0.827	0.835	0.812	0.807	0.818
250	0.837	0.831	0.842	0.820	0.816	0.825

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

**Table 8** Dataset **D**: classification performance of Ph-CNN compared to other classifiers on the external validation dataset

Task	Ph-CNN	LSVM	MLPNN	RF
UCf	0.946	0.934	0.898	0.869
UCr	0.897	0.904	0.897	0.756
Cdf	0.926	0.935	0.884	0.859
CDr	0.888	0.888	0.821	0.722
iCdf	0.931	0.943	0.905	0.863
iCDr	0.901	0.910	0.846	0.778

matrix  $CDf'_0$  by  $(CDf'_0)_i = (CDf'_i) - \mu(HS')$ . Introduce the projections  $P_{HS'} = HS'_0 \cdot (\mu(HS') - \mu(CDf'))$  and  $P_{CDf'} = CDf'_0 \cdot (\mu(HS') - \mu(CDf'))$ , then define now  $\sigma = \sqrt{\frac{\sum_{i=1}^{38} (P_{HS'})^2 + \sum_{i=1}^{60} ((P_{CDf'}) - (\mu_i(CDf') - \mu_i(HS'))^2)}{38+60}}$  and  $\mu = \frac{\mu(HS') + \mu(CDf')}{2}$ . Fix  $\alpha \in \mathbb{R}_0^+$  and define  $m_{HS} = \mu + \alpha\sigma \frac{\mu(HS')}{\|\mu(HS')\|}$  and  $m_{CDf} = \mu + \alpha\sigma \frac{\mu(CDf')}{\|\mu(CDf')\|}$ . Then, define  $HS_s^\alpha$  as the dataset collecting  $n_{HS}$  instances from a multivariate normal distribution with mean  $m_{HS}$  and covariance  $\Sigma(HS')$  and analogously  $CDf_s^\alpha$ . The two synthetic data subsets  $HS_s^\alpha$  and  $CDf_s^\alpha$  are defined by taking *ilr*-counterimages:  $HS_s^\alpha = ilr^{-1}(HS_s'^\alpha)$  and  $CDf_s^\alpha = ilr^{-1}(CDf_s'^\alpha)$ . Finally, the synthetic dataset  $D_\alpha$  is then obtained as the union  $HS_s^\alpha \cup CDf_s^\alpha$ . Setting the parameter  $\alpha$ , we provide different complexity levels in the classification task. For instance, for  $\alpha = 0$  the means of the two classes in the synthetic dataset  $D_0$  are the same, while for  $\alpha = 1$  the means of the two classes HS and Cdf are the same as in the IBD dataset; larger values

**Table 10** Dataset **D** on IBD: classification performance of Ph-CNN compared to other classifiers on Healthy vs. UCr classification task

UCr	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
60	0.445	0.375	0.517	0.509	0.221	0.384
119	0.464	0.393	0.537	0.533	0.238	0.357
178	0.444	0.372	0.520	0.519	0.328	0.449
237	0.346	0.283	0.536	0.408	0.303	0.420
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
60	0.415	0.350	0.476	0.508	0.425	0.584
119	0.528	0.463	0.596	0.455	0.387	0.525
178	0.538	0.471	0.610	0.435	0.363	0.504
237	0.489	0.417	0.557	0.400	0.337	0.463

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

of  $\alpha$  correspond to easier classification tasks. Principal component analysis of the four datasets  $D_0, D_1, D_2, D_3$  with same sample size as IBD dataset is displayed in Fig. 6.

With the same procedure, a synthetic dataset **D** is created with 10,000 samples and  $\alpha = 1$ , preserving class size ratios.

In practice, generation of the synthetic datasets was performed using the R packages *compositions* [52] and *mvtnorm* [53].

**Table 9** Dataset **D** on IBD: classification performance of Ph-CNN compared to other classifiers on Healthy vs. UCr classification task

UCf	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
63	0.659	0.604	0.709	0.510	0.449	0.573
125	0.668	0.595	0.734	0.438	0.368	0.500
188	0.650	0.599	0.707	0.541	0.438	0.604
250	0.628	0.567	0.687	0.565	0.510	0.619
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
63	0.689	0.629	0.743	0.741	0.698	0.783
125	0.644	0.582	0.703	0.742	0.690	0.792
188	0.570	0.496	0.644	0.735	0.680	0.784
250	0.606	0.547	0.667	0.760	0.707	0.816

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

**Table 11** Dataset **D** on IBD: classification performance of Ph-CNN compared to other classifiers on Healthy vs. Cdf classification task

Cdf	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.613	0.555	0.665	0.419	0.363	0.472
130	0.617	0.549	0.601	0.326	0.252	0.394
195	0.630	0.560	0.682	0.647	0.595	0.691
259	0.572	0.501	0.620	0.595	0.545	0.642
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.610	0.549	0.666	0.677	0.618	0.728
130	0.620	0.551	0.685	0.706	0.648	0.758
195	0.601	0.534	0.667	0.739	0.685	0.788
259	0.648	0.589	0.703	0.720	0.667	0.768

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

**Table 12** Dataset **D** on IBD: classification performance of Ph-CNN compared to other classifiers on Healthy vs. CDr classification task

CDr	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.241	0.172	0.311	0.138	0.073	0.198
129	0.232	0.167	0.295	0.089	0.028	0.151
193	0.202	0.131	0.273	0.169	0.101	0.236
257	0.218	0.158	0.278	0.178	0.107	0.251
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.235	0.	0.306	0.488	0.437	0.541
129	0.275	0.199	0.348	0.432	0.373	0.485
193	0.243	0.172	0.315	0.402	0.341	0.464
257	0.233	0.160	0.305	0.398	0.331	0.464

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

## Results and discussion

The  $10 \times 5$ -fold CV DAP has been applied on instances of the synthetic datasets and on the IBD datasets, comparing the performance with standard (and shallow) learning algorithms such as linear Support Vector Machines (SVM) and Random Forest (RF), and with a standard Multi Layer Perceptron (MLPNN) [54]. As expected [55], no classification task can be reliably tackled by Ph-CNN using the IBD dataset alone: the very small sample size causes the neural network to overfit after just a couple of epochs. To overcome this issue we explore the potentialities of transfer learning.

**Table 13** Dataset **D** on IBD: classification performance of Ph-CNN compared to other classifiers on Healthy vs. iCDf classification task

iCDf	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
62	0.704	0.655	0.753	0.534	0.484	0.583
124	0.702	0.642	0.760	0.414	0.346	0.482
186	0.680	0.614	0.738	0.662	0.605	0.718
247	0.681	0.614	0.739	0.561	0.507	0.621
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
62	0.679	0.622	0.739	0.787	0.746	0.831
124	0.690	0.634	0.743	0.811	0.766	0.854
186	0.685	0.630	0.742	0.791	0.741	0.836
247	0.708	0.652	0.764	0.775	0.730	0.820

The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

**Table 14** Dataset **D** on IBD: classification performance of Ph-CNN compared to other classifiers on Healthy vs. iCDr classification task

iCDr	Ph-CNN			LSVM		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
65	0.537	0.480	0.601	0.338	0.277	0.409
129	0.522	0.453	0.595	0.319	0.254	0.385
193	0.556	0.492	0.617	0.377	0.315	0.437
257	0.477	0.411	0.544	0.438	0.378	0.492
	MLPNN			RF		
$p$	MCC	min CI	max CI	MCC	min CI	max CI
63	0.526	0.475	0.581	0.552	0.492	0.612
125	0.558	0.493	0.623	0.563	0.516	0.609
188	0.459	0.388	0.527	0.566	0.516	0.616
250	0.529	0.462	0.598	0.539	0.482	0.596

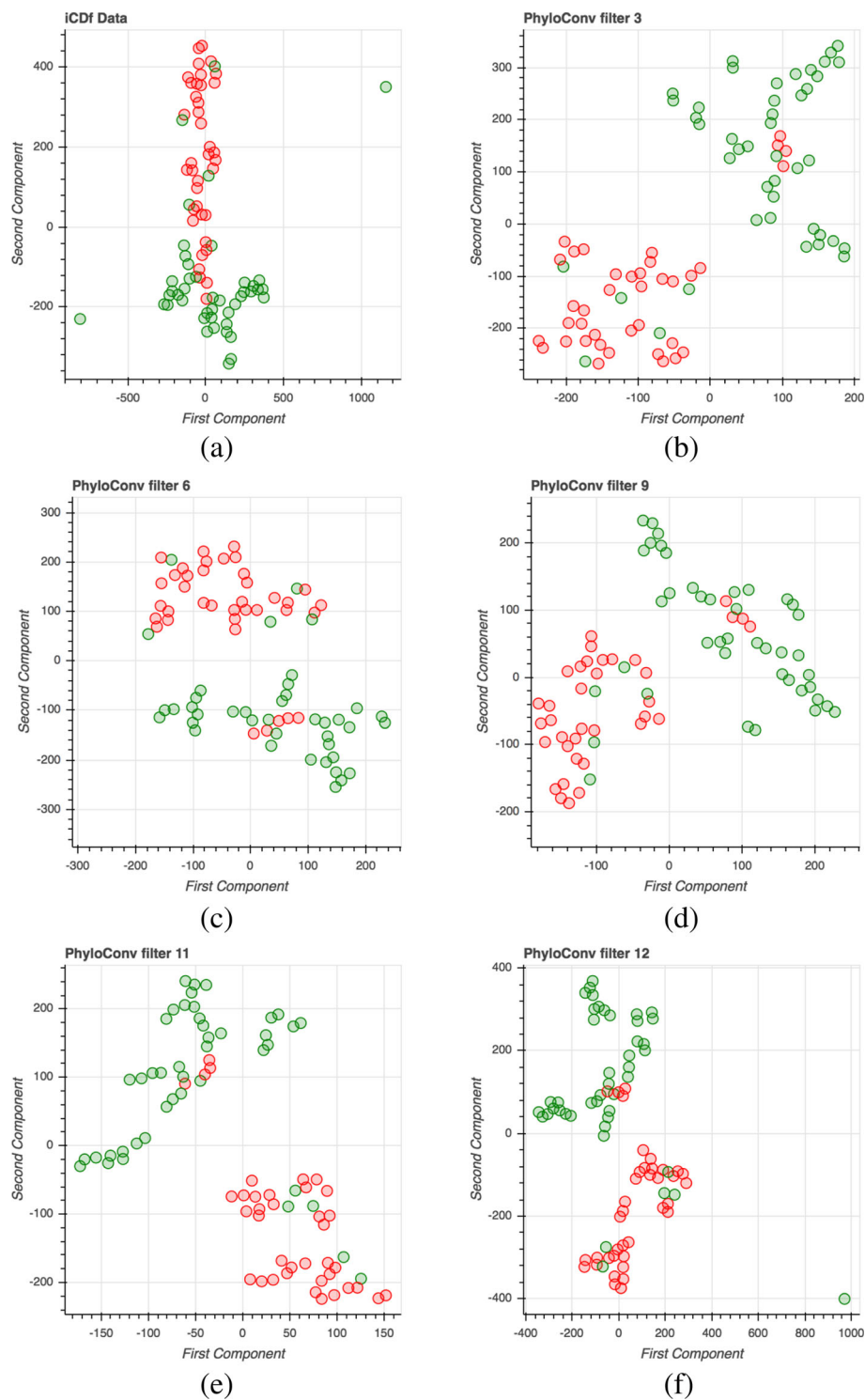
The performance measure is MCC, with 95% studentized bootstrap confidence intervals (min CI, max CI). Models are computed for  $p = \{25\%, 50\%, 75\% \text{ and } 100\%\}$  of total number of features for each task. Comparing algorithms are linear Support Vector Machines (LSVM), Random Forest (RF) and MultiLayer Perceptron (MLPNN)

As a first experiment, we apply the DAP on **D**. In this case, the SELU activation function is used for every layer. The results of the Ph-CNN DAP on **D** are listed in Tables 2, 3, 4, 5, 6, 7 (internal validation) and Table 8 (external validation) on the six classification tasks Healthy vs. {UCf, UCr, Cdf, CDr, iCDf and iCDr}; MCC on DAP internal validation is shown with 95% studentized bootstrap confidence intervals [56].

The second experiment is based on a domain adaptation strategy. The Ph-CNN is first trained on the synthetic dataset **D**, then all layer but the last one are frozen, the last layer is substituted by a 2-neurons Dense layer and then retrained on the IBD dataset. Since only the last layer is trained in the second step, the term domain adaptation is best describing the methodology rather than the more generic transfer learning. Here, the activation function is the ReLU for every layer. The results of the Ph-CNN DAP together with the comparing classifiers are listed in Tables 9, 10, 11, 12, 13, 14 (internal validation) and Table 15 (external validation).

**Table 15** Dataset **D** on IBD: classification performance of Ph-CNN compared to other classifiers on the external validation dataset

Task	Ph-CNN	LSVM	MLPNN	RF
UCf	0.741	0.740	0.666	0.699
UCr	0.583	0.497	0.608	0.678
Cdf	0.858	0.642	0.705	0.707
CDr	0.853	0.654	0.654	0.597
iCDf	0.842	0.418	0.401	0.920
iCDr	0.628	0.414	0.414	0.418



**Fig. 7** t-SNE projections of the original features at initial layer (subfigure **a**) and after 3, 6, 9, 11, 12 convolutional filters (subfigures **b-f**). Green for healthy subjects, red for iCDf patients

As an observation, Ph-CNN tends to misclassify more the samples in class Healthy, rather than those in the other class, for each classification task. In

Fig. 7 we show the embeddings of the original features at 6 different levels (after initial input and after 5 PhyloConv filters) for the iCDf task (IBD dataset)

by projecting them in two dimensions via t-distributed Stochastic Neighbor Embedding (t-SNE) [57] with perplexity = 5 and 5,000 iterations. While at input level the problem seems hardly separable, the classes tend to form distinct clusters during the flow through convolutional filters applied on OTUs close in the taxonomy.

**Computational details** The Ph-CNN is implemented as a Keras v2.0.8 layer, while the whole DAP is written in Python/Scikit-Learn [58]. All computations were run on a Microsoft Azure platform with 2x NVIDIA Tesla K80 GPUs.

## Conclusions

We introduced here Ph-CNN, a novel DL approach for the classification of metagenomics data exploiting the hierarchical structure of the OTUs inherited by the corresponding phylogenetic tree. In particular, the tree structure is used throughout the prediction phase to define the concept of OTU neighbours, used in the convolution process by the CNN. Results are promising, both in terms of learning performance and biomarkers detection. Extensions of the Ph-CNN architecture are addressing the testing of different tree distances, optimization of neighbours detection and of the number of Phylo-Conv layers. Further, different feature selection algorithms, either generic or DL-specific can be adopted [59–61]. Improvements are expected on the transfer learning and domain adaptation procedures, such as learning on synthetic data and testing on metagenomics, and applying to larger datasets. Finally, beyond the metagenomics applications, we observe that Ph-CNN is a general purpose algorithm, whose use can be extended to other data for which the concept of nearest features can be defined. This is true for all data types that are metrizable, i.e. whenever an embedding exists of the features into a metric space. As an example, we are currently investigating the transcriptomics case, where a grounded distance between genes can be defined by mixing the data-independent Gene Ontology semantic similarity with the correlation between gene expression in the studied dataset [62] through a dedicated multilayer network structure. From a general perspective, the metagenomics and transcriptomics case represent just the first steps towards a more general strategy for effectively exploiting the potential of CNNs, especially for omics data.

## Abbreviations

CD: Crohn's disease; CNN: Convolutional neural networks; DAP: Data analysis protocol; IBD: Inflammatory bowel disease; MCC: Matthews correlation coefficient; MDS: MultiDimensional scaling; MLPNN: Multi-Layer perceptron; OTU: Operational taxonomic unit; RF: Random forest; SVM: Support vector machine; UC: Ulcerative colitis

## Acknowledgements

The authors wish to thank Alessandro Zandonà for help during the analyses of the IBD dataset and Calogero Zarbo for help in a first implementation of the approach.

## Funding

Publication costs for this article were covered by FBK CI003 institutional funds.

## Availability of data and materials

Data and code are available at <https://gitlab.fbk.eu/MPBA/phylogenetic-cnn>.

## About this supplement

This article has been published as part of BMC Bioinformatics Volume 19 Supplement 2, 2018: Proceedings of Bringing Maths to Life (BMTL) 2017. The full contents of the supplement are available online at <https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-19-supplement-2>.

## Authors' contributions

YG, VM and DF implemented and performed all computational work; MC supervised the biological aspects of the projects; CA supervised the mathematical aspects of the project; CA and GJ conceived the project; CF and GJ directed the project and drafted the manuscript. All authors discussed the results, read and approved the final manuscript.

## Ethics approval and consent to participate

Not applicable.

## Consent for publication

Not applicable.

## Competing interests

The authors declare that they have no competing interests.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Author details

<sup>1</sup>Fondazione Bruno Kessler (FBK), Via Sommarive 18 Povo, I-38123 Trento, Italy. <sup>2</sup>Max Planck Institute for Intelligent Systems, Spemannstraße 34, 72076 Tübingen, Germany. <sup>3</sup>Centre for Medical Informatics, Usher Institute, University of Edinburgh, 9 Little France Road, EH16 4UX Edinburgh, UK. <sup>4</sup>Department of Mathematics, University of Trento, Via Sommarive 14 Povo, I-38123 Trento, Italy.

Published: 8 March 2018

## References

- Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, et al. Opportunities And Obstacles For Deep Learning In Biology And Medicine. *BioRxiv*. 2017;142760.
- Mamoshina P, Vieira A, Putin E, Zhavoronkov A. Applications of Deep Learning in Biomedicine. *Mol Pharm*. 2016;13(5):1445–54.
- Chaudhary K, Poirion OB, Lu L, Garmire L. Deep Learning based multi-omics integration robustly predicts survival in liver cancer. *BioRxiv*. 2017;114892.
- Zacharakis EI. Prediction of protein function using a deep convolutional neural network ensemble. *PeerJ Comput Sci*. 2017;3:e124.
- Min S, Lee B, Yoon S. Deep learning in bioinformatics. *Brief Bioinform*. 2016;18(5):542.
- Ditzler G, Polikar R, Rosen G. Multi-Layer and Recursive Neural Networks for Metagenomic Classification. *IEEE Trans NanoBioscience*. 2015;14(6):608–16.
- Arango-Argoty GA, Garner E, Pruden A, Heath LS, Vikesland P, Zhang L. DeepARG: A deep learning approach for predicting antibiotic resistance genes from metagenomic data. *BioRxiv*. 2017;149328.
- Fang H, Huang C, Zhao H, Deng M. gCoda: Conditional Dependence Network Inference for Compositional Data. *J Comput Biol*. 2017;24(7):699–708.
- Qin J, Li R, Raes J, Arumugam M, Burgdorf KS, Manichanh C, et al. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*. 2010;464(7285):59–65.

10. Illumina. Inc. 16S Metagenomics Studies with the MiSeq System Illumina, Inc.; 2017 Application Note: Microbial Genomics. [https://www.illumina.com/content/dam/illumina-marketing/documents/products/appnotes/appnote\\_16s\\_sequencing.pdf](https://www.illumina.com/content/dam/illumina-marketing/documents/products/appnotes/appnote_16s_sequencing.pdf). Accessed Oct 2017.
11. Albanese D, De Filippo C, Cavalieri D, Donati C. Explaining Diversity in Metagenomic Datasets by Phylogenetic-Based Feature Weighting. *PLoS Comput Biol*. 2015;11(3):e1004186.
12. Fukuyama J, Rumker L, Sankaran K, Jeganathan P, Dethlefsen L, Relman DA, et al. Multidomain analyses of a longitudinal human microbiome intestinal cleanout perturbation experiment. *PLoS Comput Biol*. 2017;13(8):e1005706.
13. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86(11):2278–324.
14. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. *Advances in Neural Information Processing Systems* vol. 25. Red Hook: Curran Associates, Inc; 2012. p. 1097–105.
15. Stuessy TF, König C. Patrocladistic classification. *Taxonomy*. 2008;57(2): 594–601.
16. Cox TF, Cox MAA. *Multidimensional Scaling*. Boca Raton: Chapman and Hall; 2001.
17. de Vienne DM, Aguilera G, Ollier S. Euclidean nature of phylogenetic distance matrices. *Syst Biol*. 2011;60(6):826–32.
18. Shen H, Huang JZ. Sparse principal component analysis via regularized low rank matrix approximation. *J Multivar Anal*. 2007;99:1015–34.
19. Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc*. 2001;96:1348–60.
20. Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, Droge J, et al. Critical Assessment of Metagenome Interpretation – a benchmark of metagenomics software. *Nat Methods*. 2017;14:1063–1071.
21. Chollet F, Keras. 2015. <https://github.com/fchollet/keras>. Accessed Oct 2017.
22. The MicroArray Quality Control (MAQC) Consortium. The MAQC-II Project: A comprehensive study of common practices for the development and validation of microarray-based predictive models. *Nat Biotechnol*. 2010;28(8):827–38.
23. The SEQC/MAQC-III Consortium. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nat Biotechnol*. 2014;32(9):903–14.
24. Jurman G, Riccadonna S, Visintainer R, Furlanello C. Algebraic Comparison of Partial Lists in Bioinformatics. *PLoS ONE*. 2012;7(5):e36540.
25. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta Protein Struct*. 1975;405(2):442–51.
26. Baldi P, Brunak S, Chauvin Y, Andersen CAF, Nielsen H. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*. 2000;16(5):412–24.
27. Jurman G, Riccadonna S, Furlanello C. A comparison of MCC and CEN error measures in multi-class prediction. *PLoS ONE*. 2012;7(8):e41882.
28. Sokol H, Leducq V, Aschard H, Pham HP, Jegou S, Landman C, et al. Fungal microbiota dysbiosis in IBD. *Gut*. 2017;66(6):1039–48.
29. Alberti A, Poulain J, Engelen S, Labadie K, Romac S, Ferrera I, et al. Viral to metazoan marine plankton nucleotide sequences from the Tara Oceans expedition. *Sci Data*. 2017;4:170093.
30. Giarratano Y. *Phylogenetic Convolutional Neural Networks in Metagenomics*. M.Sc. Thesis, University of Trento, Italy. 2016.
31. St John K. Review Paper: The Shape of Phylogenetic Treespace. *Syst Biol*. 2017;66(1):e83–e94.
32. Entringer RC. Distance in graphs: trees. *JCMCC. J Comb Math Comb Comput*. 1997;24:65–84.
33. Klambauer G, Unterthiner T, Mayr A, Hochreiter S. Self-Normalizing Neural Networks. 2017. [ArXiv:1706.02515](https://arxiv.org/abs/1706.02515).
34. Kingma D, Ba J. Adam A Method for Stochastic Optimization. 2014. [ArXiv:1412.6980](https://arxiv.org/abs/1412.6980). 3rd International Conference for Learning Representations, ICLR. 2015.
35. Gorodkin J. Comparing two K-category assignments by a K-category correlation coefficient. *Comput Biol Chem*. 2004;28:367–74.
36. De Borda M. Mémoire sur les élections au scrutin. *Hist de l'Académie Royale des Sci*. 1781;1781:657–64.
37. Saari DG. *Chaotic Elections! A Mathematician Looks at Voting*. Providence: AMS; 2001.
38. Jurman G, Merler S, Barla A, Paoli S, Galea A, Furlanello C. Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics*. 2008;24(2):258–64.
39. Morgan XC, Tickle TL, Sokol H, Gevers D, Devaney KL, Ward DV, et al. Dysfunction of the intestinal microbiome in inflammatory bowel disease and treatment. *Genome Biol*. 2012;13(9):R79.
40. Sokol H, Pigneur B, Watterlot L, Lakhdari O, Bermúdez-Humarán LG, Gratadoux JJ, et al. *Faecalibacterium prausnitzii* is an anti-inflammatory commensal bacterium identified by gut microbiota analysis of Crohn disease patients. *Proc Natl Acad Sci*. 2008;105(43):16731–6.
41. Kuczynski J, Stombaugh J, Walters WA, González A, Caporaso JG, Knight R. 10. In: *Using QIIME to analyze 16s rRNA gene sequences from microbial communities*. Hoboken: John Wiley & Sons, Inc.; 2005. p. Unit 10.7.
42. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, et al. QIIME allows analysis of high-throughput community sequencing data. *Nat Methods*. 2010;7(5):335–6.
43. Edgar RC. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*. 2010;26(19):2460–1.
44. McDonald D, Price MN, Goodrich J, Nawrocki EP, DeSantis JT, Probst A, et al. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J*. 2012;6(3):610–8.
45. DeSantis JT, Hugenholtz P, Keller K, Brodie EL, Larsen N, Piceno YM, et al. NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes. *Nucleic Acids Res*. 2006;34(suppl 2):W394–W399.
46. Caporaso JG, Bittinger K, Bushman FD, DeSantis JT, Andersen GL, Knight R. PyNAST: a flexible tool for aligning sequences to a template alignment. *Bioinformatics*. 2009;26(2):266.
47. Stamatakis A. RAxML version 8 a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*. 2014;30(9):1312–3.
48. Price MN, Dehal PS, Arkin AP. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*. 2010;5(3):e9490.
49. Pruesse E, Quast C, Knittel K, Fuchs BM, Ludwig W, Peplies J, et al. SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB. *Nucleic Acids Res*. 2007;35(21):7188–96.
50. Aitchison J. *The Statistical Analysis of Compositional Data*. London - New York: Chapman and Hall; 1986.
51. Egozcue JJ, Pawłowsky-Glahn V, Mateu-Figueras G, Barceló-Vidal C. Isometric logratio transformations for compositional data analysis. *Math Geol*. 2003;35(3):279–300.
52. van den Boogaart KG, Tolosana-Delgado R. “compositions”: a unified R package to analyze Compositional Data. *Comput Geosci*. 2008;34(4):320–38.
53. Mi X, Miwa T, Hothorn T. mvtnorm: New numerical algorithm for multivariate normal probabilities R J. 2009;1(1):37–9.
54. Bishop CM. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press; 1995.
55. Angermueller C, Pärnamäa T, Parts L, Stegle O. Deep learning for computational biology. *Mol Syst Biol*. 2016;12(7):878.
56. DiCiccio TJ, B E. Bootstrap confidence intervals (with Discussion). *Stat Sci*. 1996;11:189–228.
57. Maaten Lvd, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res*. 2008;9(Nov):2579–605.
58. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. *J Mach Learn Res*. 2011;12: 2825–30.
59. Nezhada MZ, Zhub D, Lib X, Yanga K, Levy P. SAFS: A Deep Feature Selection Approach for Precision Medicine. 2017. [ArXiv 1704.05960](https://arxiv.org/abs/1704.05960).
60. Roy D, Murty KSR, Mohan CK. Feature selection using Deep Neural Networks. In: *International Joint Conference on Neural Networks (IJCNN)*. Red Hook: IEEE; 2015. p. 1–6.
61. Li Y, Chih-Yu C, Wasserman WW. Deep Feature Selection: Theory and Application to Identify Enhancers and Promoters. *J Comput Biol*. 2016;23(5):322–36.
62. Jurman G, Maggio V, Fioravanti D, Giarratano Y, Landi I, et al. Convolutional neural networks for structured omics: OmicsCNN and the OmicsConv layer. 2017. [ArXiv 1710.05918](https://arxiv.org/abs/1710.05918).